**2. Elaborazione – Iterazione 1**

**Introduzione**

In questa fase si passa all’elaborazione del progetto allo scopo di implementare, iterazione dopo iterazione, il sistema nelle sue singole parti, attraverso in processo di tipo incrementale. In particolare, lo scopo della prima iterazione è quello di implementare i soli scenari di successo del caso d’uso *UC1: Ricerca volo* e del caso d’uso *UC2: Gestisci prenotazione*. Inoltre, viene allestito un caso d’uso di start up al fine di testare gli elementi prodotti nella corrente iterazione.

**Aggiornamento caso d’uso UC2: Gestisci prenotazione**

Il caso d’uso *UC2: Gestisci prenotazione* viene modificato rispetto a quanto già definito in fase di ideazione per renderlo conforme al tipo di operazioni eseguite dal codice e al tipo di collaborazione tra le classi software di questa iterazione. Con le modifiche apportate il caso d’uso diventa il seguente:

|  |  |
| --- | --- |
|  |  |
| **Nome del caso d’uso** | **UC2: Gestisci prenotazione** |
| **Portata** | Applicazione Finch Airlines |
| **Livello** | Obiettivo utente |
| **Attore primario** | Cliente |
| **Parti interessate e Interessi** | * Cliente: vuole gestire una prenotazione creandola, modificando i dati di una effettuata in precedenza o cancellandola. |
| **Pre-condizioni** | Il cliente ha effettuato la ricerca di un volo |
| **Garanzia di successo** | Il sistema ha effettuato il salvataggio delle informazioni della prenotazione |
| **Scenario principale di successo** | 1. Il cliente vuole effettuare una prenotazione e seleziona i voli desiderati tra quelli proposti dal sistema 2. Il cliente seleziona un posto e il tipo desiderato per il singolo volo 3. Il cliente seleziona il tipo di bagaglio desiderato per il singolo volo 4. Il sistema visualizza un riepilogo dei dati della prenotazione 5. Il cliente conferma la prenotazione 6. Il sistema spedisce una e-mail riepilogativa dell’ordine |
| **Estensioni** | \*a. In qualsiasi momento il sistema fallisce   1. Il sistema avvisa il cliente dell’errore 2. Il sistema ritorna alla schermata di ricerca iniziale   1b. Il cliente vuole modificare una prenotazione esistente   1. Il cliente inserisce il numero di prenotazione. 2. Il sistema ricerca la prenotazione e la mostra al cliente   2a. Il sistema non trova la prenotazione   1. Il sistema avvisa il cliente 2. Il cliente inserisce un nuovo numero di prenotazione   3. Il cliente seleziona l’operazione da svolgere sulla prenotazione:  3a. Il cliente seleziona la modifica dei dati anagrafici   1. Il cliente modifica i dati (nome, cognome, documento identificativo) 2. Il cliente conferma la modifica dei dati 3. Il sistema conferma la modifica effettuata   3b. Il cliente seleziona l’acquisto di un posto a sedere   1. Il cliente seleziona il posto tra quelli ancora disponibili 2. Il sistema aggiorna il totale corrente 3. Il cliente conferma la scelta   3c. Il cliente seleziona l’acquisto di un bagaglio   1. Il cliente sceglie la quantità di bagagli da aggiungere alla prenotazione 2. Il sistema aggiorna il totale corrente 3. Il cliente conferma la scelta   4. Il cliente conferma la modifica della prenotazione  5. Il sistema mostra il totale e procede con l’eventuale pagamento  1c. Il cliente vuole annullare una prenotazione esistente   1. Il cliente inserisce il numero di prenotazione 2. Il sistema ricerca la prenotazione e la mostra al cliente   2a. Il sistema non trova la prenotazione   1. Il sistema avvisa il cliente 2. Il cliente inserisce un nuovo numero di prenotazione   3. Il cliente effettua la rimozione della prenotazione  4. Il sistema effettua lo storno  1d. Il cliente vuole cercare e visualizzare una prenotazione   1. Il cliente inserisce il numero di prenotazione 2. Il sistema ricerca la prenotazione e la mostra al cliente   2a. Il sistema non trova la prenotazione   1. Il sistema avvisa il cliente 2. Il cliente inserisce un nuovo numero di prenotazione   3. Il cliente visualizza la prenotazione |
| **Requisiti speciali** | Non presenti |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Frequenza elevata nella creazione delle nuove prenotazioni |
| **Varie** |  |

**Analisi Orientata agli oggetti**

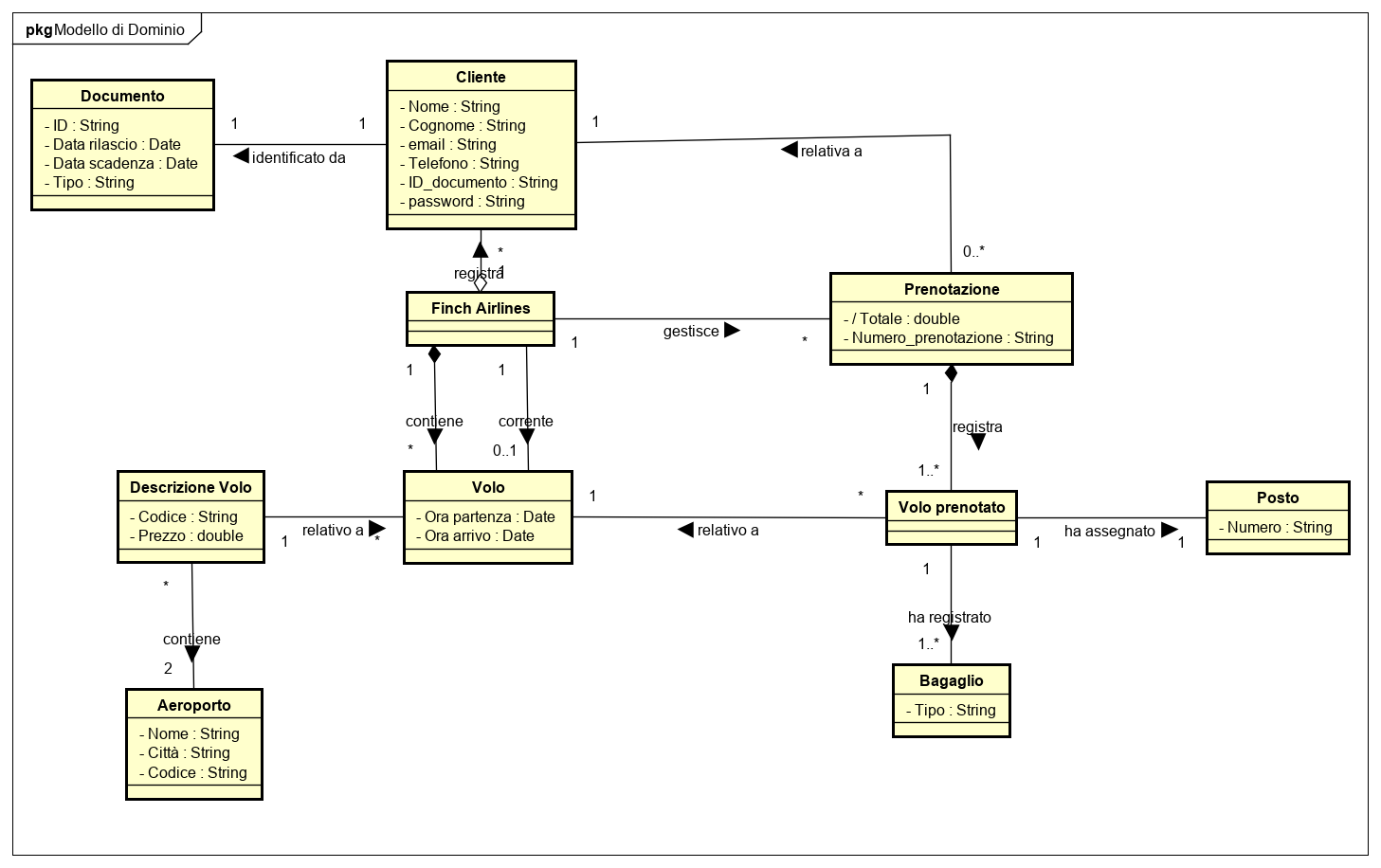
Vengono qui di seguito rappresentati gli schemi realizzati in fase di Analisi mediante approccio di tipo Object Oriented. In particolare: Modello di Dominio, SSD.

Modello di Dominio

La fase di analisi ha portato alla redazione del Modello di Dominio, facente parte della disciplina della Modellazione del Business, contenente le seguenti classi concettuali:

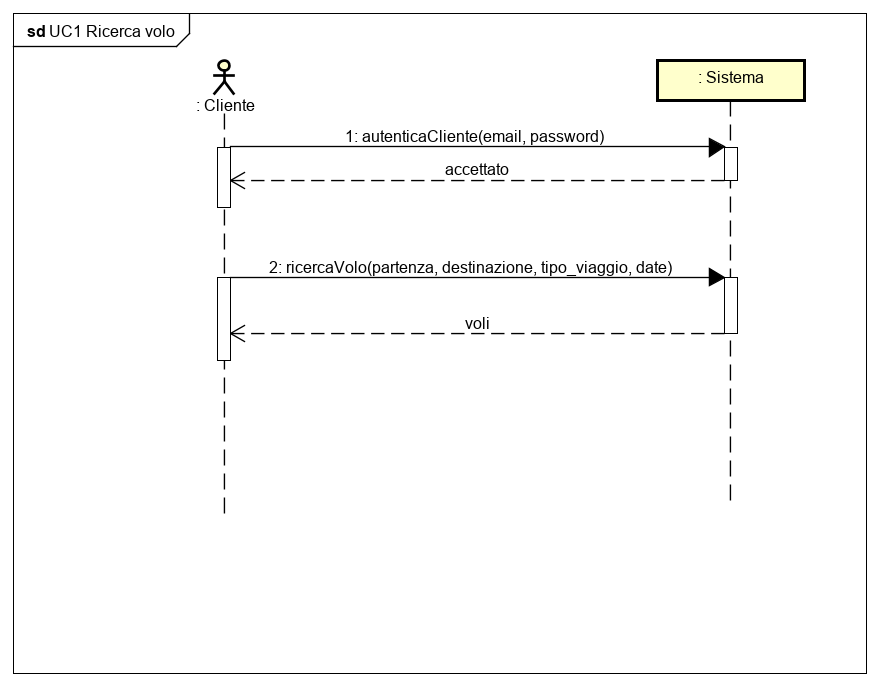
* Finch Airlines: rappresenta il sistema
* Cliente: rappresenta il cliente che vuole effettuare una prenotazione (per sé stesso in questa iterazione)
* Documento: rappresenta il documento identificativo utilizzato dal Cliente per prenotare
* Prenotazione: rappresenta la prenotazione contenente l’insieme dei dati dei singoli voli che la costituiscono
* Volo prenotato: rappresenta il volo associato ad una particolare prenotazione e l’insieme dei servizi ad esso correlati
* Posto: rappresenta il posto associato ad un volo e le relative informazioni descrittive
* Bagaglio: rappresenta il bagaglio associato ad un volo e le relative informazioni descrittive
* Volo: rappresenta un volo operato dalla compagnia aerea
* Descrizione Volo: rappresenta la descrizione associata ad un particolare volo
* Aeroporto: rappresenta un aeroporto e le relative informazioni descrittive

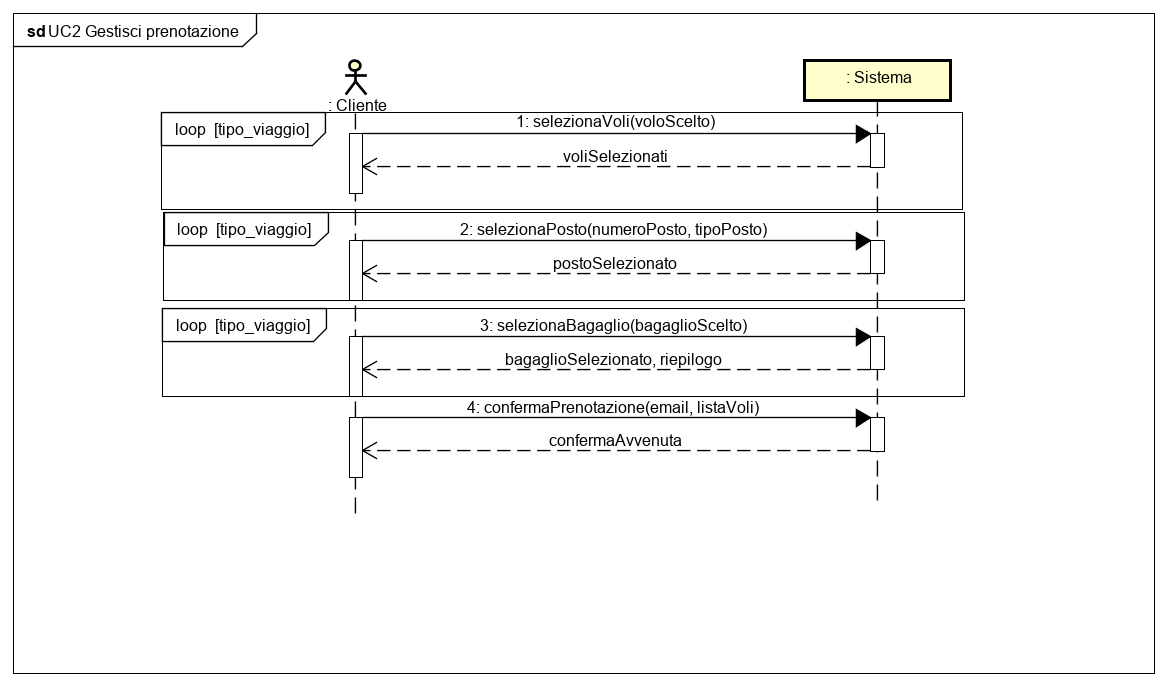
Il Modello di Dominio è il seguente:



Diagrammi di sequenza di sistema

Per l’iterazione corrente viene prodotto un ulteriore documento, relativo alla disciplina dei Casi d’Uso, che è il Diagrammi di Sequenza di Sistema (SSD). In particolare, vengono realizzati quelli per i casi d’uso *UC1: Ricerca volo* e *UC2: Gestisci prenotazione*, ovvero:



**

Contratti delle operazioni

In questa sezione vengono descritte le principali operazioni di sistema del caso d’uso UC2 mediante i contratti delle operazioni.

**Contratto CO1: selezionaVoli**

**Operazione:** *selezionaVoli(voloScelto:* Volo*, indice*: int*)*

**Riferimenti:** caso d’uso: Gestisci prenotazione

**Pre-condizioni:** È stata effettuata una ricerca di voli

**Post-condizioni:**

* È stata creata un’istanza *listaVoli[]* di *VoloPrenotato*
* *listaVoli[i]* è stato associato a *volo*

**Contratto CO2: selezionaPosto**

**Operazione:** *selezionaPosto(numeroPosto:* String, *tipoPosto:* String, *indice*: int*)*

**Riferimenti:** caso d’uso: Gestisci prenotazione

**Pre-condizioni:** Sono stati selezionati dei voli

**Post-condizioni:**

* È stata creata un’istanza *posto* di *Posto*
* *posto* è stato associato a *listaVoli[i]*
* Gli attributi *numeroPosto* e *tipoPosto* sono stati aggiornati con *numeroPosto* e *tipoPosto*

**Contratto CO3: selezionaBagaglio**

**Operazione:** *selezionaBagaglio (bagaglioScelto:* String, *indice*: int*)*

**Riferimenti:** caso d’uso: Gestisci prenotazione

**Pre-condizioni:** Sono stati selezionati i volie posti

**Post-condizioni:**

* È stata creata un’istanza *bagaglio* di *Bagaglio*
* *bagaglio* è stato associato a *listaVoli[i]*
* L’attributo *bagaglioScelto* è stato aggiornato con *bagaglioScelto*

**Contratto CO4: confermaPrenotazione**

**Operazione:** *confermaPrenotazione (cliente:* Cliente, *listaVoli*: VoloPrenotato[]*)*

**Riferimenti:** caso d’uso: Gestisci prenotazione

**Pre-condizioni:** Sono stati selezionati i voli**,** posti e bagagli

**Post-condizioni:**

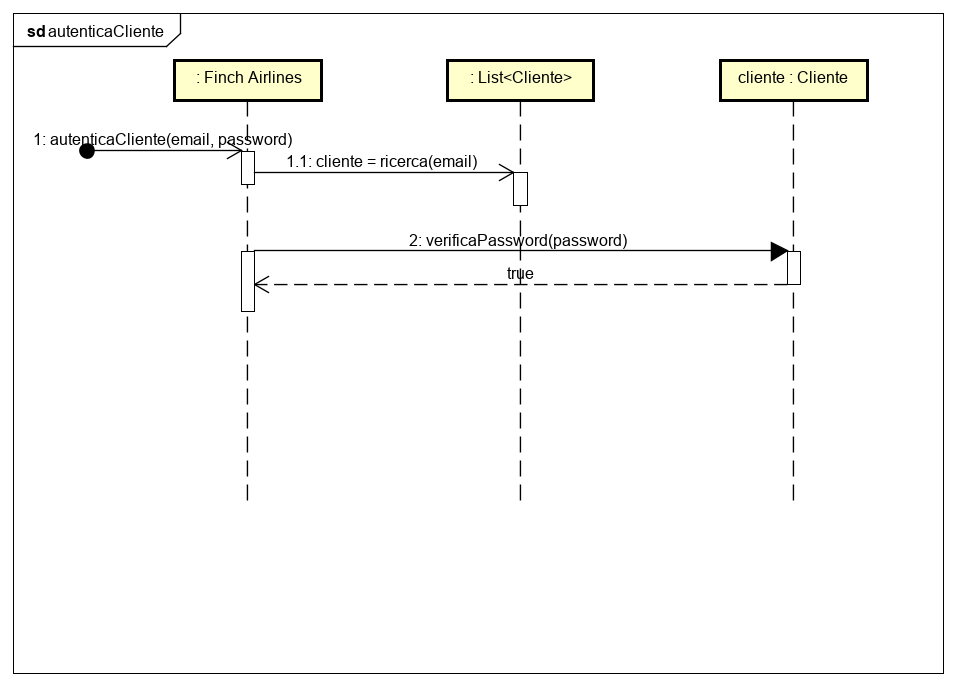
* È stata creata un’istanza *prenotazione* di *Prenotazione*
* *prenotazione* è stata associata a *finchAirlines*
* *cliente* è stato associato a *prenotazione*
* *listaVoli[]* è stato associato a *prenotazione*
* L’attributo *numeroPrenotazione* di *prenotazione* è stato aggiornato
* L’attributo *totale* di *prenotazione* è stato aggiornato

**Progettazione Orientata agli Oggetti**

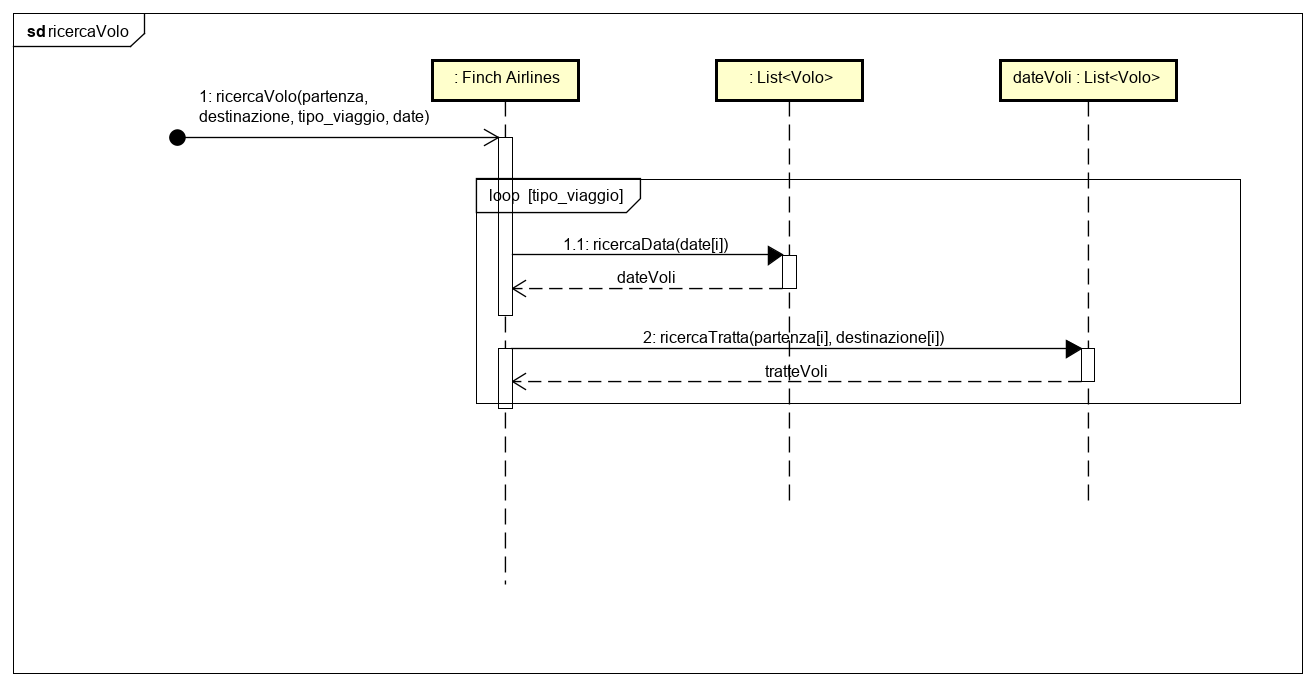
Un’ulteriore fase eseguita all’interno della corrente elaborazione è quella progettazione, al fine di ottenere due approcci, quello dinamico (diagrammi di sequenza) e quello statico (diagramma delle classi DCD). Inoltre, il principio seguito in tale fase richiede una progettazione guidata dalle responsabilità, dove quindi si fa uso dei pattern GRASP per l’implementazione dei diagrammi sopra citati. I diagrammi rappresentati fanno riferimento ai casi d’uso *UC1: Ricerca volo* e *UC2: Gestisci prenotazione* in termini dinamici (diagrammi di sequenza) e in termini statici (DCD).

Diagrammi di sequenza

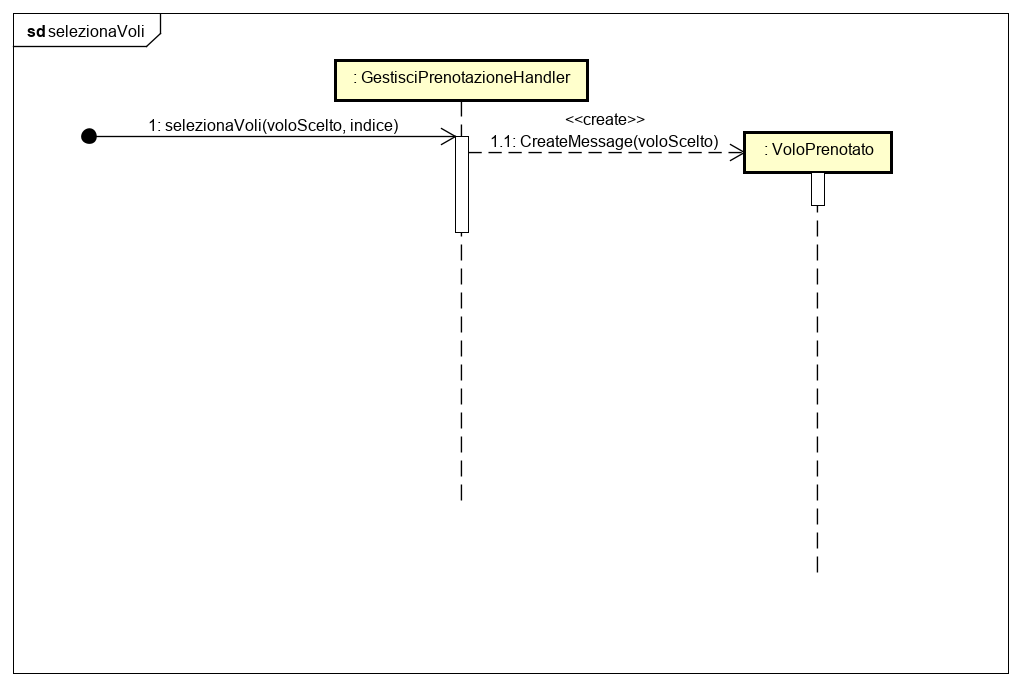
* Autenticazione del Cliente all’interno del sistema



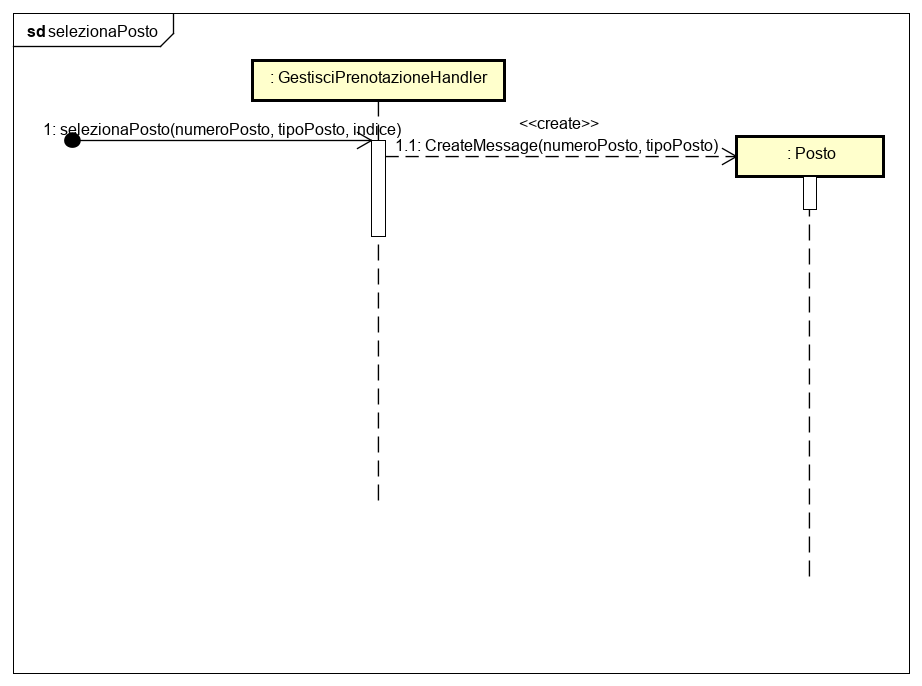
* Ricerca di voli date le informazioni di interesse per il cliente



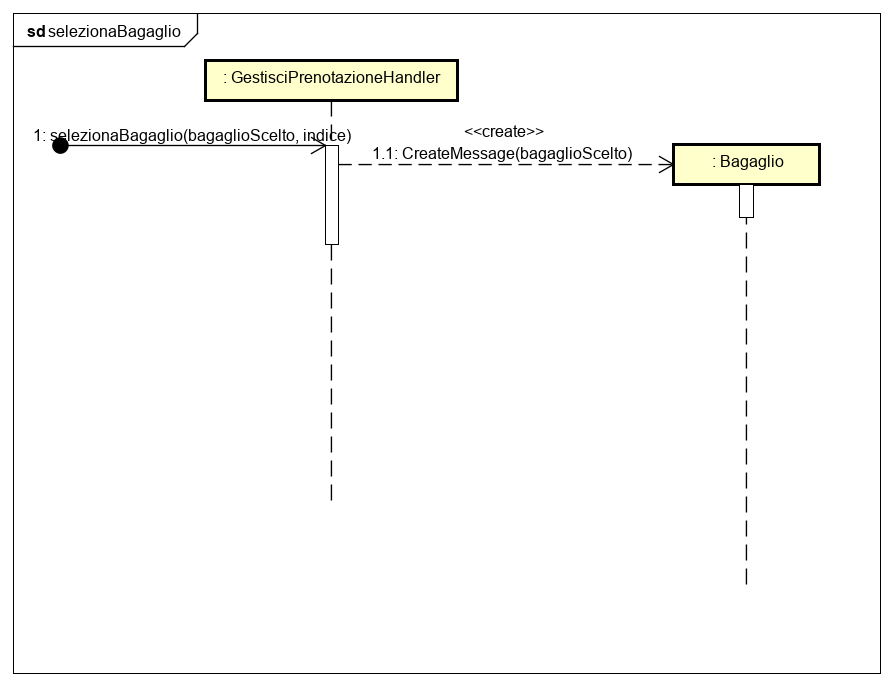
* Selezione del volo per la singola tratta prenotata dal cliente



* Selezione del posto per il volo corrente della prenotazione



* Selezione del bagaglio per il volo corrente della prenotazione



* Conferma e creazione della prenotazione effettuata dal cliente

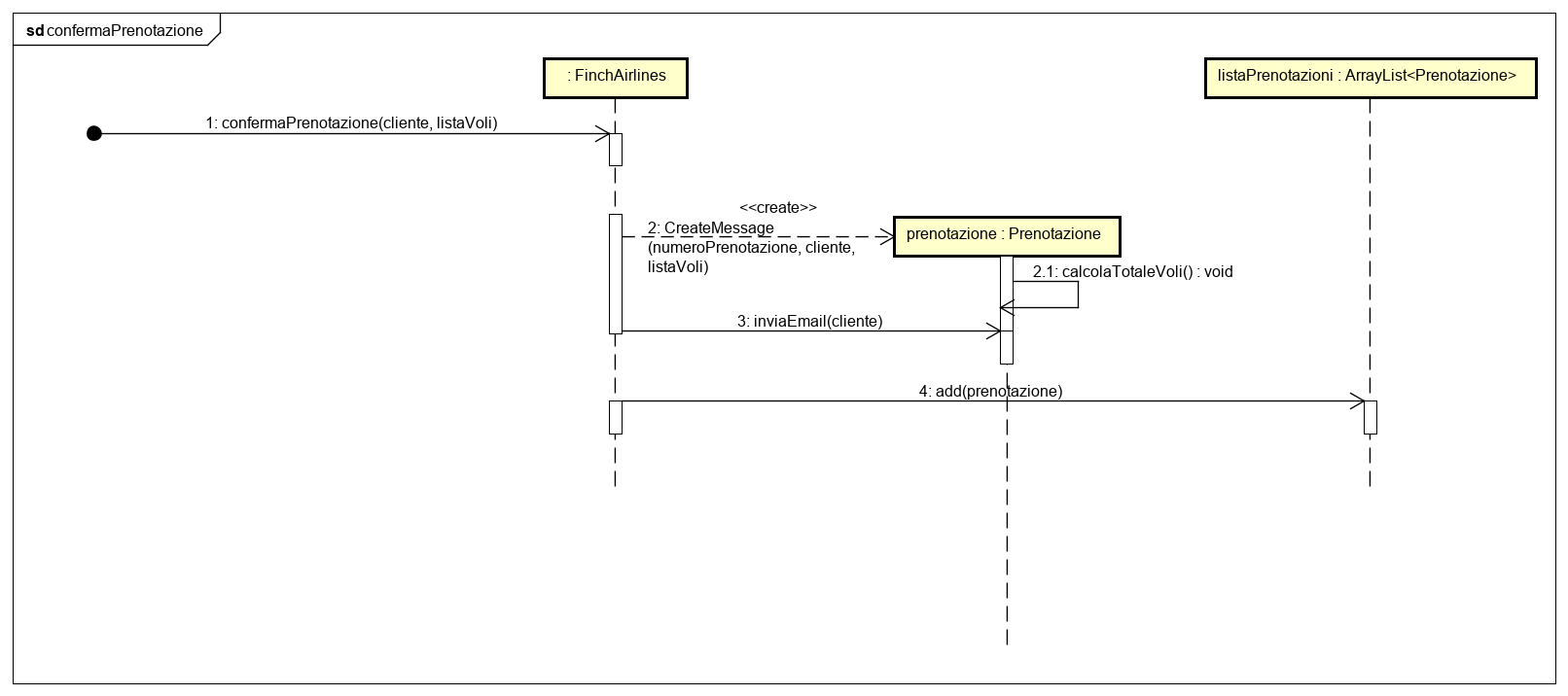


Diagramma delle classi

